IN THE SPECIFICATION

Please replace the paragraph at page 28, line 20, to page 29, line 4, with the following rewritten paragraph:

Initially, the communication section 132 of the backend apparatus ~~330~~ 430 receives information specifying rn, products distribution information pd and tag output information $G(s_{k,i})$ which are transmitted from the reader 120 (step S50). Information specifying rn, the products distribution information pd and the tag output information $G(s_{k,i})$ which have been received are stored in the memory 136a. Then the controller 136 enters 1 for n, and stores it in the memory 136a (step S51). The controller 136 then causes a hash calculator 433 to extract a confidential value $s_{n,1}$ from the database memory 131 while referring to values of n and j in the memory 136a (step S52), and causes the hash function H to be applied thereto rn times and also causes the hash function G to be applied subsequently, thus allowing a hash value $G(H^j(s_{n,1}))$ (j=rn) to be calculated (step S53).

Please replace the paragraph at page 32, lines 7-24, with the following rewritten paragraph:

Initially, the backend apparatus 530 receives products distribution information pd and tag output information $E_{KG}(s_{k,i})$ transmitted by the reader 120 by the communication section 132 (step S70). Received products distribution information pd and tag output information $E_{KG}(s_{k,i})$ are stored in the memory 136a. Then, the controller 136 enters 1 for n, and stores it in the memory 136a (step S71). The controller 136 then causes the encrypted function calculator 533 (equivalent to "third calculator") to extract the confidential value $s_{n,1}$ from the database 131 while referring to the value n in the memory 136a (step S72). The controller 136 then enters 0 for j, and stores it in the memory 136a (step S73). The controller 136 causes the encrypted function calculator 533 to calculate an encrypted text $E_{KG}(E^j_{KH}(s_{n,1}))$ (equivalent to “result of a calculation in the third calculator”) while referring to the value of j in the memory 136a (step S74). It should be noted that $E^j_{KH}(s_{n,1})$ implies applying a common key encryption function E to the confidential value $s_{n,1}$ j times using the common key KH. Then the comparator 134 acquires the encrypted text $E_{KG}(E^j_{KH}(s_{n,1}))$ from the hash calculator 133 and acquires ~~tag output information~~ tag output information $E_{KG}(s_{k,i})$ from the memory 136a, and compare them against each other (step S75).

Please replace the paragraph at page 45, lines 3-11, with the following rewritten paragraph:

[0083] Combinations of generated initial elements ($f_{1,0}, \ldots, f_{u,0}, \ldots, f_{d,0}$) (equivalent to “combinations each comprising $d(d \geq 2)$ elements $e_{u,vu}$ ($u \in \{1, \cdots, d\}$) and corresponding to respective tag ID information id_k” where vu represents an integer equal to or greater than 0 and indicating the number of times the element $e_{u,vu}$ is updated and the suffix vu of the element $e_{u,vu}$ represents v_u) are stored in the confidential value memory 811 of respective allotted tag devices 810. In the description to follow, a combination of initial elements which is stored in the confidential value memory 811 of each tag device 810 is indicated by ($e_{1,0}, \ldots,$ ~~eu, 0, …, ed, 0~~ $e_{u,0}, \ldots, e_{d,0}$).

Please replace the paragraph at page 45, lines 19-24, with the following rewritten paragraph:

[0085] As shown in Fig. 16 A, one set of combinations of initial elements 811a (($e_{1,0},$ ~~e2, 0~~ $e_{2,0}$)=($b_{1,2,0},$ ~~b2, 2, 0~~ $b_{2,2,0}$)) which corresponds to the tag ID information id is stored in the confidential value memory 811 of the tag device 810. It is to be noted that part of the element $e_{u,vu}$ which is stored in the confidential value memory 811 is also stored in the confidential value memory of another tag device as a corresponding element in another tag device.

Please replace the paragraph at page 61, line 22, to page 62, line 8, with the following rewritten paragraph:

An embodiment 11 is a modification of the embodiment 10, and differs from the embodiment 10 in that a manifold value z_u which assumes t_u kinds ($t_u \geq 2$) of values is stored for each u ($u \in \{1, \cdots, d\}$) in a manifold value memory of a tag device, and a tag output information $a_{k,i} = G(e_{1,v1} \mid z_1 \mid \cdots \mid e_{d,vd} \mid z_d)$ for a bit combination value of each element $e_{u,vu}$ extracted from a confidential value memory and either one of manifold values z_u is used as an output value. In addition, while the update of each element $e_{u,vu}$ in the confidential value memory which corresponds to each u ($u \in \{1, \cdots, d\}$) is performed once for commutations of t times, in the embodiment 11, the point in time of communication when element $e_{u,vu}$ is updated is shifted, so that either one of elements $e_{u',vu'}$ ($u' \in \{1, \cdots, d\}$) in the confidential value memory is updated each time the tag device delivers the tag output information $a_{k,i}$. This prevents the tag device from being traced if the tag device is tampered with at any point in time of communication.

Please replace the paragraph at page 63, lines 6-18, with the following rewritten paragraph:

[0124] The manifold value generator 1115 can be illustrated by a counter which counts $z_u = 1 \cdots t_u$ for each u ($u \in \{1, \cdots, d\}$), a hash calculator which performs a calculation of z_u=H(seed, x_u), $x_u \in \{1, \cdots, t_u\}$ or a hash calculator which performs a calculation of z_u~~=H^{*}(seed)~~ H^{xu}(seed), $x_u \in \{1, \cdots, t_u\}$. In the description to follow, manifold value z_u is expressed as $z_u=\pi_u(x_u)$, $x_u \in \{1, \cdots, t_u\}$. Preferably, π_u is set up such that for an equal value of u, manifold values $z_u=\pi_u(x_u)$ corresponding to $x_u \in \{1, \cdots, t_u\}$ do not coincide.

Please replace the paragraph at page 65, lines 18-20, with the following rewritten paragraph:

A distinction of the embodiment 11 over the embodiment 10 resides in that the processing indicated at ~~step S26~~ Fig. 26 is performed in place of processings at steps S208 ~ S213 shown in Fig. 23.

Please replace the paragraph at page 80, lines 23-27, with the following rewritten paragraph:

[0159] Information such as the privileged ID information (sid_h) or the like is received by the communication section ~~62~~ 2062 of the security server 2060 (its input is accepted) (step S305), and is sent to a read/write section 2064. This also triggers the random number generator ~~63~~ 2063 (equivalent to "random value generator") to generate a random number r_h' (step S306).

Please replace the paragraph at page 88, lines 15-20, with the following rewritten paragraph:

[0173] Information including the privileged ID information sid_h or the like is received by the communication section 2062 of the security server 2260 (step S344), and the first encrypted text $epk_j(id_h \mid r)$ which constitutes the privileged ID information sid_h is sent to the ID extractor ~~266~~ 2266 while the key ID information kid_j is sent to the read/write section 2064. The key ID information kid_j is also recorded in the memory 2065a.

Please replace the paragraph at page 103, line 24, to page 104, line 12, with the following rewritten paragraph:

Initially, the controller ~~23~~ 2023 determines whether or not there existed a given trigger (opportunity) to update the privileged ID information (step S412). What can be cited as such a trigger are that the privileged ID information has been read from the tag device 2610, that a count indicating a number of times the privileged ID information within the tag device 2610 has been used has reached a given value or the like. In the absence of a given trigger, a determination rendered at step S412 is continued, and in the presence of a given trigger, the read/write section 2624 (equivalent to "privileged ID extractor") extracts one privileged ID information sid_h-j from the privileged ID memory 2625 (step S413). The selection of this one privileged ID information sid_h-j may take place at random, or may be in the sequence of an array in the manner of sid_h-1, sid_h-2, ··· and returning to sid_h-1 again after sid_h-p. The extracted one privileged ID information sid_h-j is sent from the read/write section 2624 to the interface 2022 (equivalent to "privileged ID output section"), and thence transmitted (delivered) to the tag device 2610 (step S414).

Please replace the paragraph at page 113, lines 4-6, with the following rewritten paragraph:

Initially, a decrypting processing of privileged ID information which is performed when demanding a backend apparatus ~~50~~ 3050 to acquire information relating to ID will be described.

Please replace the paragraph at page 113, line 26, to page 114, line 17, with the following rewritten paragraph:

[0219] The transmitted privileged ID information sid_h and key ID information kid_j are received by a communication section 3072 (equivalent to "privileged ID input section") of the security server 3070 (accepted as inputs) (step S505), and the privileged ID information sid_h is fed to a decryptor ~~74~~ 3074 (equivalent to "ID calculator") while the key ID information kid_j is fed to a read/write section 3073. The read/write section 3073 (equivalent to "key extractor") extracts a secret key sk_j which corresponds to the key ID information kid_j which is sent thereto from a key memory 3071, and sends it to the decryptor 3074 (step S506). The decryptor 3074 calculates a tag ID information id_h which is decrypted from the privileged ID information sid_h using the privileged ID information sid_h and the secret key sk_j which are sent thereto. In this example, the tag ID information id_h is calculated by a calculation of $id_h=(id_h \cdot pk_j^r)/(g^r)^{skj}$. It is to be noted that the index "skj" in this calculation formula means "sk_j". The calculated tag ID information id_h is sent to the communication section 3072, which then transmits it toward the client apparatus 3020 through the network 3080 (step S508). The client apparatus 3020 receives the transmitted tag ID information id_h at its communication section 3021 (step S509), and utilizes this tag ID information id_h for a subsequent inquiry to the backend apparatus 3050.